

WE CLAIM:

1. A method for representing field structures in a markup language document, comprising:
 - determining properties corresponding to a field that relates to at least one section of an application document;
 - mapping the properties of the field into at least one of a markup language element, an attribute, and a value; and
 - storing the properties of the field in the markup language document.
2. The method of Claim 1, further comprising determining whether the field is one of a complex field and a simple field.
3. The method of Claim 2, wherein an instruction portion of the field comprises at least one of richly formatted text and embedded additional fields when the field is a complex field.
4. The method of Claim 2, wherein an instruction portion of the field excludes richly formatted text and embedded additional fields when the field is a simple field.
5. The method of Claim 2, further comprising representing the field with a fldSimple element when the field is determined to be a simple field.
6. The method of Claim 2, further comprising representing the field with at least one of a fldChar element and instrText element when the field is determined to be a complex field.
7. The method of Claim 1, further comprising:
 - determining properties corresponding to an additional field that relates to at least one section of the application document;

mapping the properties of the additional field into at least one of a markup language element, an attribute, and a value; and

storing the properties of the additional field in the markup language document.

8. The method of Claim 1, further comprising:

determining whether properties associated with all fields of the application document have been stored in the markup language document; and

processing further fields when the properties associated with all fields have not been stored in the markup language document.

9. The method of Claim 1, wherein the properties of the fields stored in the markup language document are understood by an application that understands the markup language when the field is not native to the application.

10. The method of Claim 1, wherein the markup language document is manipulated on a server to substantially reproduce the field of the application document notwithstanding the presence of an application that generated the markup language document.

11. A computer-readable medium for representing fields in a markup language document, comprising:

determining properties relating to one or more fields used within a word-processing document;

determining whether the field is one of a complex field and a simple field;

writing the properties into at least one of a markup language element, an attribute, and a value; and

storing the properties in the markup language document such that the fields of the word-processing document are substantially maintained when the markup language document is parsed by an application.

12. The computer-readable medium of Claim 11, wherein the properties of the fields stored in the markup language document are understood by an application that understands the markup language when the field is not native to the application.

13. The computer-readable medium of Claim 11, wherein the markup language document is manipulated on a server to substantially reproduce the field of the word-processing document notwithstanding the presence of an application that generated the markup language document.

14. The computer-readable medium of Claim 11, wherein an instruction portion of the field comprises at least one of richly formatted text and embedded additional fields when the field is a complex field.

15. The computer-readable medium of Claim 11, wherein an instruction portion of the field excludes richly formatted text and embedded additional fields when the field is a simple field.

16. The computer-readable medium of Claim 11, further comprising representing the field with a fldSimple element when the field is determined to be a simple field.

17. The computer-readable medium of Claim 11, further comprising representing the field with at least one of a fldChar element and instrText element when the field is determined to be a complex field.

18. The computer-readable medium of Claim 11, further comprising:
determining properties corresponding to an additional field that relates to at least one section of the word-processing document;
determining whether the additional field is one of a complex field and a simple field;

mapping the properties of the additional field into at least one of a markup language element, an attribute, and a value; and

storing the properties of the additional field in the markup language document.

19. A system for representing fields in a markup language document, comprising:

an application that is configured to:

determine properties relating to a field included in at least one section of an application document;

determine whether the field is one of a complex field and a simple field;

map the properties into at least one of a markup language element, an attribute, and a value; and

store the properties in the markup language document; and

a validation engine configured to validate the markup language document.

20. The system of Claim 19, wherein the properties of the fields stored in the markup language document are understood by an additional application that understands the markup language when the field is not native to the additional application.

21. The system of Claim 19, wherein an instruction portion of the field comprises at least one of richly formatted text and embedded additional fields when the field is a complex field.

22. The system of Claim 19, wherein an instruction portion of the field excludes richly formatted text and embedded additional fields when the field is a simple field.

23. The system of Claim 19, wherein the markup language document is manipulated on a server to generate the field of the application document notwithstanding the presence of the application that generated the markup language document.